IPA Better Life with IT

障害分析手法解説

IPA/SECセミナー
「失敗から学ぶ組み込みソフト高信頼化のためのアプローチ法紹介」
～2014年度 製品・制御システム高信頼化部会成果報告～
2015年5月18日

IPA/SEC　障害事例検証ＷＧ委員
パイオニアシステムテクノロジー株式会社
津田 昌之

© 2015 IPA, All Rights Reserved

組み込みシステムで「高品質なモノづくり」が出来る理由

SEC Software Reliability Enhancement Center

- （理由1）品質を確保・管理・保障する仕組みがある。

  例えば・・・

  - 社内のソフトウェア開発工程内に設計レビューがある。
  - 設計レビュー記録を残す仕組みがある。
  - 設計レビューでの指摘事項が解決されていることをチェックする仕組みがある。
  - 出荷前に出荷可否判定する仕組みがある。

- （理由2）開発対象ドメインを熟知し、**品質を作りこむための経験・ノウハウ**を持った人がいる。

© 2015 IPA, All Rights Reserved

「高品質なモノづくり」をどうやって維持するか

SEC
Software Reliability Enhancement Center

1. ベテラン技術者の**経験・ノウハウ**を次世代に残る形で整理する。

2. 若い技術者にその**経験・ノウハウ**を知りたいと思わせる。

3. 若い技術者にその**経験・ノウハウ**を身に付けてもらう。

4. **ノウハウ**を獲得する仕組みを整備して継続実行する。

© 2015 IPA, All Rights Reserved

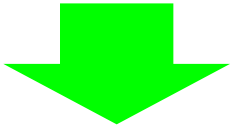
ノウハウはどこから/どうやって集めるか

SEC
Software Reliability Enhancement Center

■ どうやってノウハウを手に入れるか

- （ベテラン技術者がいるうちは・・・）先輩・上司のレビュー
- 他人の失敗から学ぶ　⇒　教訓の利活用
- 自らの失敗から学ぶ　⇒　障害事例分析の実施

そもそも失敗が許されない環境となっているのでは？



数少ない失敗事例からより多くの学びを得るために

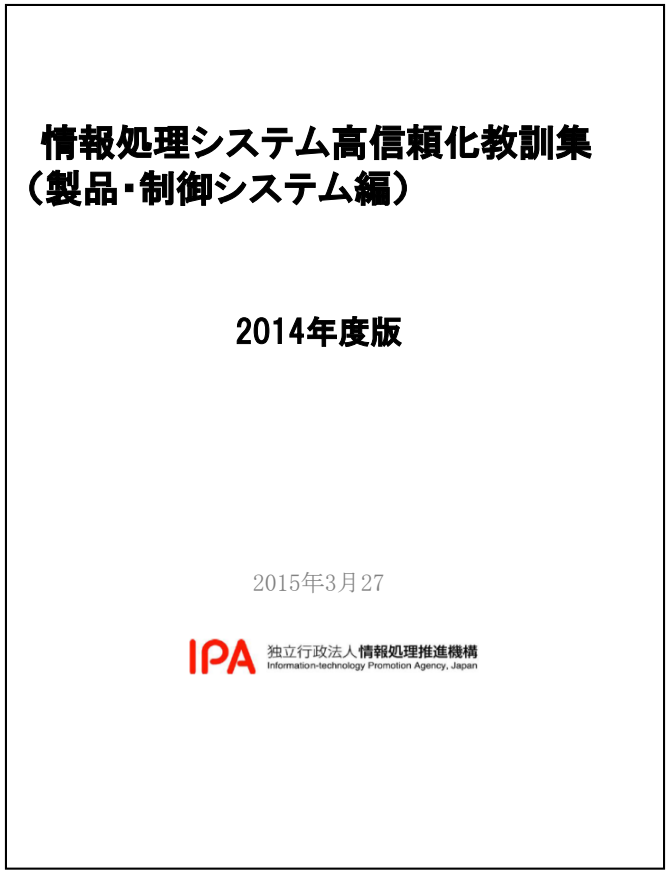
「失敗から学ぶ手法」を習得する必要がある。
（障害分析手法）

© 2015 IPA, All Rights Reserved

2014年度のアウトプット

SEC Software Reliability Enhancement Center

- 情報処理システム高信頼化教訓集（製品・制御システム編） 2014年度版



2015年3月27日
IPA・ソフトウェア高信頼化センターWEBに公開。

New!

2014年度版では「PART Ⅳ 障害分析手法事例解説書」を追加。

© 2015 IPA, All Rights Reserved

SEC Software Reliability Enhancement Center

目次

1. はじめに
2. 障害分析手法と作業
3. 障害分析事例解説
   3.1 未来都市モノレール障害の分析
   3.2 未来都市モノレール障害のなぜなぜ分析事例
   3.3 堰堤洪水吐ゲート異常作動の分析事例 1
   3.4 堰堤洪水吐ゲート異常作動の分析事例 2
4. 再発防止活動の事例
   4.1 A社の再発防止活動事例
   4.2 B社の再発防止活動事例

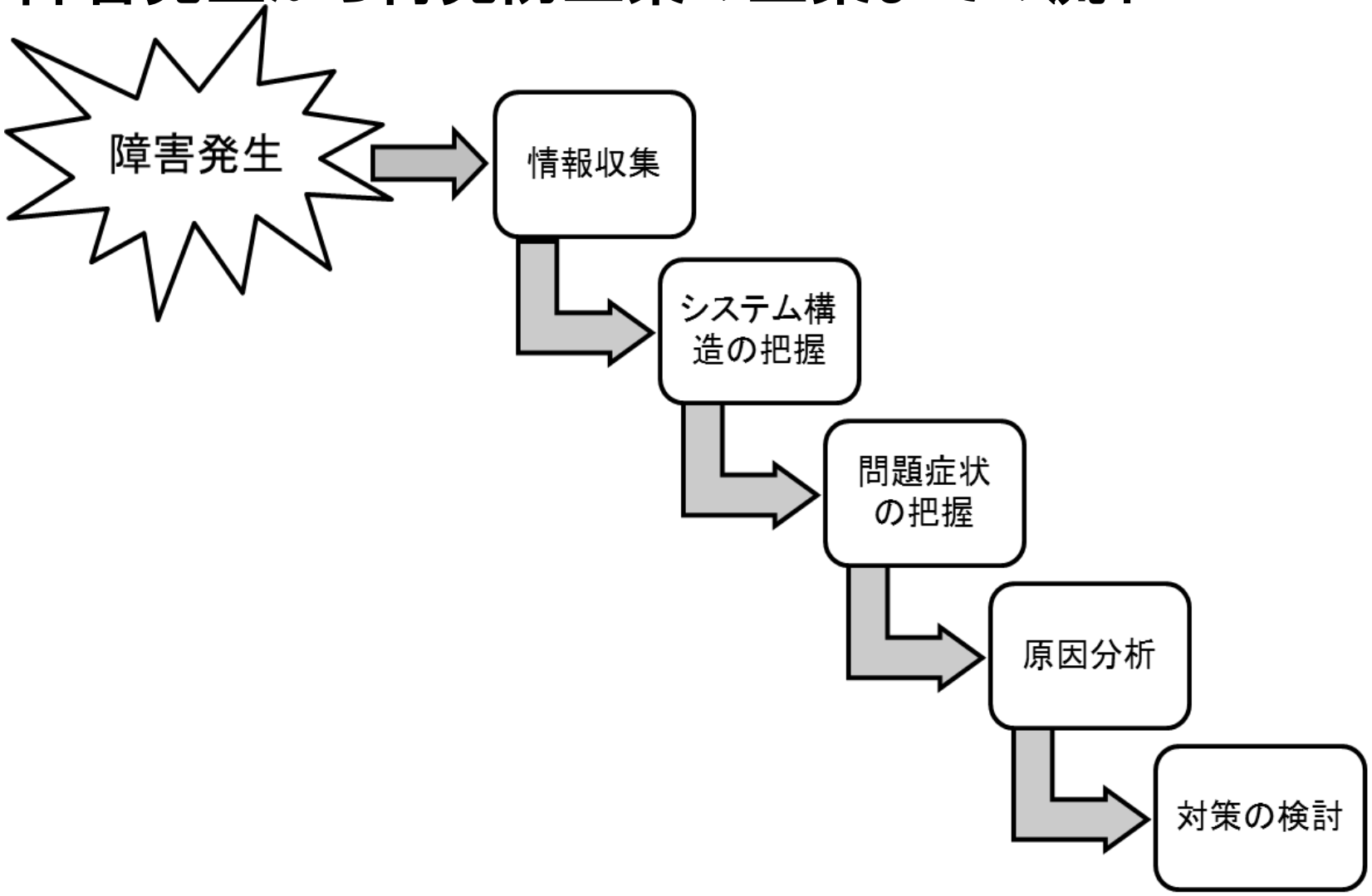
障害分析手法と分析作業について、架空の障害事例を用いて解説

© 2015 IPA, All Rights Reserved

障害分析の流れ

SEC
Software Reliability Enhancement Center

■ 障害発生から再発防止策の立案までの流れ



© 2015 IPA, All Rights Reserved

障害分析手法と分析作業の概覧

SEC
Software Reliability Enhancement Center

タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	問題症状の把握	事故経過表、VTA
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

青字:PART Ⅲで紹介している手法

© 2015 IPA, All Rights Reserved

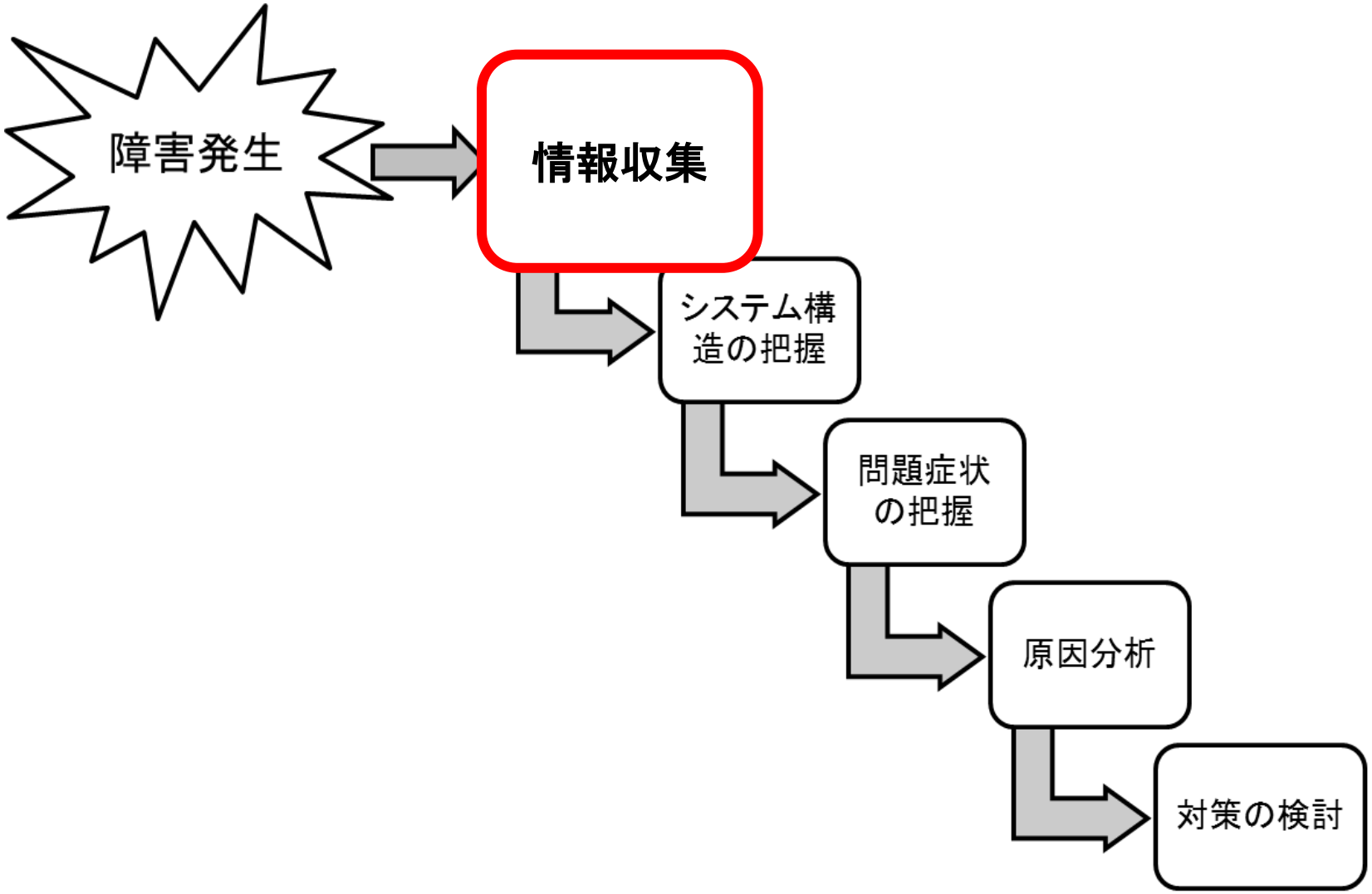
未来都市モノレール障害の分析手法解説

「未来都市モノレール」障害事例※を用いて、障害発生から再発防止策の立案までの手順に沿って、障害分析手法や分析の考え方について解説します。

※ 実際に発生した障害を参考にしていますが、架空のものです。

© 2015 IPA, All Rights Reserved

1. 情報収集



© 2015 IPA, All Rights Reserved

情報収集の手法

タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	問題症状の把握	事故経過表、VTA
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

© 2015 IPA, All Rights Reserved

障害の分析に際してまず行うのが情報収集。

以下の3つの視点で情報を収集する。

- 何で発生したのか（モノ）
- 登場人物は誰か（ヒト）
- どんな状況でどんなことが起きたのか（コト）

© 2015 IPA, All Rights Reserved

目的に合わせた視点(ヒト・モノ・コト)の使い分け

SEC
Software Reliability Enhancement Center

再発防止 未然防止	不具合修正	責任追及
仕事の改善	物の改善	個人・組織の改善
“コト”中心の視点	“モノ”中心の視点	“ヒト”中心の視点
根本原因の特定	直接原因の特定	責任の所在の特定
なぜなぜ分析	FTA、FMEA (なになに分析)	(だれだれ分析)

© 2015 IPA, All Rights Reserved

SEC Software Reliability Enhancement Center

障害の分析に必要となる情報の例

- 障害の発生状況(関係者や目撃者へのヒアリング結果など)
- 組み込みシステムの稼動情報(ログデータなど)
- 障害の再現手順(ユーザーへのヒアリング、取扱説明書など)
- 外部環境(温度、湿度、天候、明るさ、速度、路面状況、・・・と、それらの変化)
- システム構成(システムブロック図など)

※システムの種類によって必要な情報は異なる。

© 2015 IPA, All Rights Reserved

情報の整理

SEC
Software Reliability Enhancement Center

収集した情報は大きく分けて次の2つに分類できる

- 発生した事象(コト)や関係する人物(ヒト)など、障害発生の経緯に関する情報
- 組み込みシステムの動作・構造(モノ)に関する情報

© 2015 IPA, All Rights Reserved

情報整理のポイント

SEC Software Reliability Enhancement Center

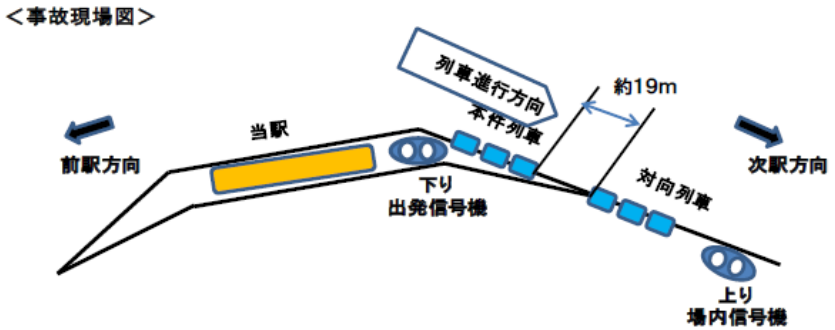
☞ 作業のポイント

- ヒアリング結果は、「事実」と「事実以外（感情、意見、推測）」を分ける。
- 定量データは、収集されたときの状況、出所、データ収集・管理の仕方なども調べ、データの信頼性を確認できるようにする。
- 事実情報は時系列に並べて整理する。

  ⇒　原因となった事象は必ず結果の前に起きている

© 2015 IPA, All Rights Reserved

情報収集と整理の具体例　障害発生の経緯

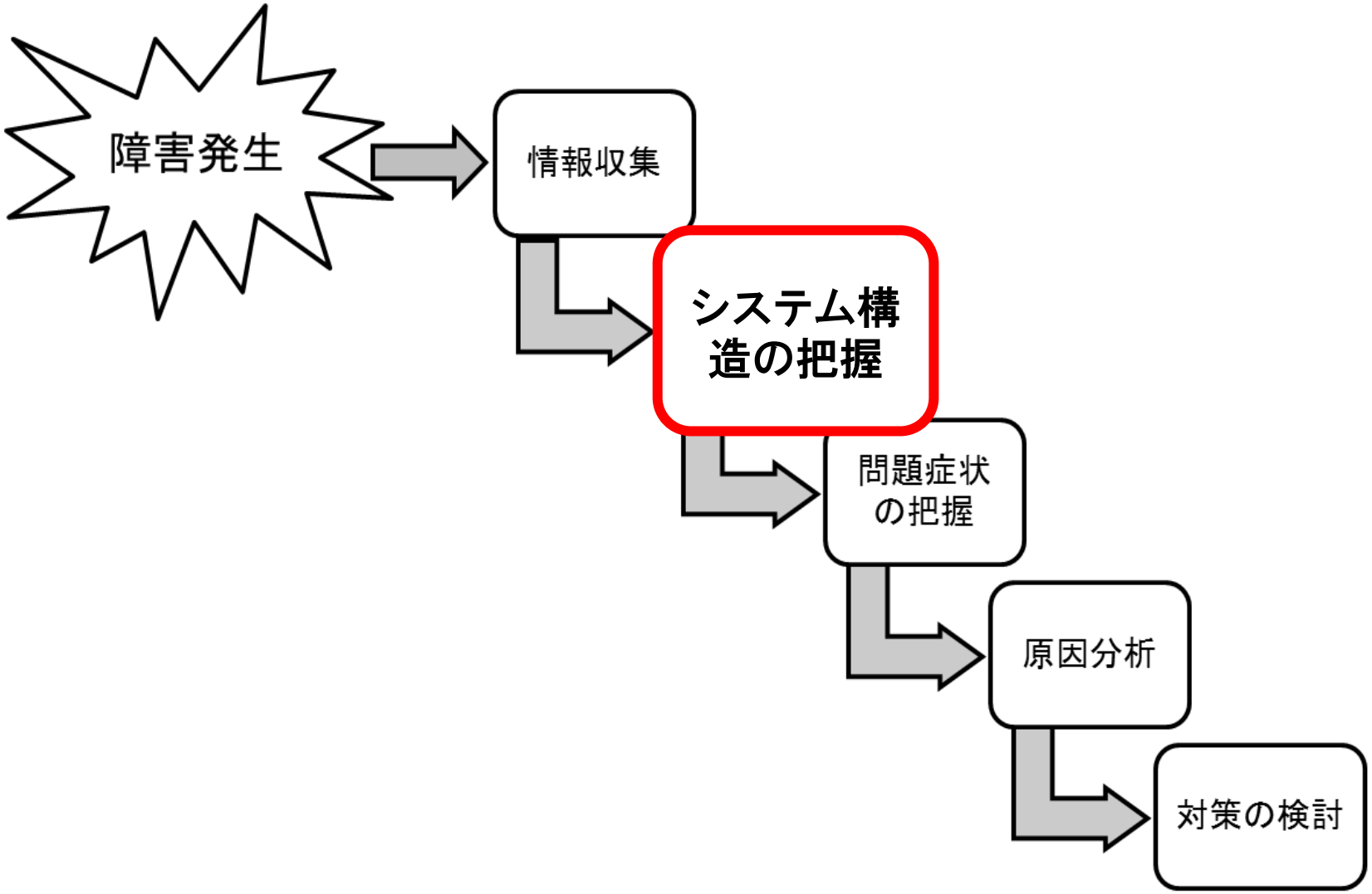


本事例では事故の状況を詳しく知るため、情報収集の手法のうち「関係者へのヒアリング」と「実地調査」を使用している。

未来都市モノレール株式会社の下り列車は、平成26年3月31日、前駅を定刻（9時50分）に出発した。
列車は同駅を出発する際に急加速し、その後、運転士がワンハンドルマスコンを力行位置としていないのにもかかわらず加速する状態となった。
本件運転士は車両の異常に気付いたが、すぐに列車を止めようとは思わなかった。最終的には非常ブレーキを使用すれば止まるかもしれないという思いがあったのかもしれない。
列車は当駅進入時にブレーキ不足の状態となり、運転士は非常ブレーキと保安ブレーキを使用したが所定位置に停止せず、その先の分岐器に衝突し停止した。
一方、当駅で下り列車とすれ違う予定であった上り列車の運転士は下り列車が冒進してくるのを認めたため、非常ブレーキを使用し、下り列車の約19m手前で停止した。
下り列車の車両及び分岐器等の物損が生じたが、死傷者はなかった。

© 2015 IPA, All Rights Reserved

2. システム構造の把握



© 2015 IPA, All Rights Reserved

システム構造把握の手法

SEC
Software Reliability Enhancement Center

タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	問題症状の把握	事故経過表、VTA
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

© 2015 IPA, All Rights Reserved

システム構造(モノ)の把握

障害が発生した組み込みシステムの機能ブロックを構成要素とするシステムの全体像や、構成要素間の関連を把握するために図や表を作成する。

- 既存の資料があれば参考にする。
- 自分の担当範囲に原因があるのではないかと感じてる人が作成開始。(待ちの姿勢は時間を浪費するだけ)
- 自分の担当範囲を中心にその周囲について、関係者にヒアリングしながらまとめ、自分の理解が正しいことを確認する。(知らないうちに変わっていた・・・。ということがないか、念のため確認)

© 2015 IPA, All Rights Reserved

システム構造把握のポイント

SEC
Software Reliability
Enhancement Center

☞ 作業のポイント

- システム全体図は存在しないのが当たり前と考え、既存の資料を探すことに時間をかけない。
- 自分で書いてみることで、システムへの理解を深める。
- 自分の担当外の部分も含め、システムを俯瞰することで障害分析の視野を広げる。

© 2015 IPA, All Rights Reserved

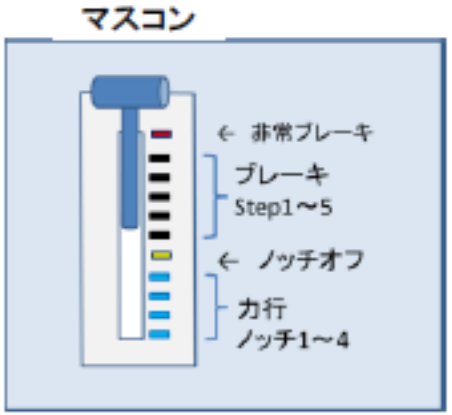
システム構造把握の具体例①　主回路

3両編成全車が電動車で、走行用に12台の主電動機（モーター）が搭載されている。
本件編成のVVVFインバータは、1両目である101号と3両目である102号に搭載され、それぞれ自車の主電動機4台と中間車である201号の主電動機2台の計6台を制御している。
「VVVFインバータ」とは、電圧及び周波数ともに変えることが可能なインバータ（直流を交流に変換する装置）をいう。

本事例では、経験のないドメインの障害分析だったため、あまり手の込んだ手法を用いず、「図」と「表」で構造を理解した。

© 2015 IPA, All Rights Reserved

システム構造把握の具体例②　制御回路



力行ノッチと加圧される指令線の関係

		マスコン設定							
		後進（上り方向）				前進（下り方向）			
		1	2	3	4	1	2	3	4
指令線番号	2	○	○	○	○	－	－	－	－
	3	－	－	－	－	○	○	○	○
	4	－	○	○	○	－	○	○	○
	5	－	－	○	○	－	－	○	○
	8	－	－	－	○	－	－	－	○

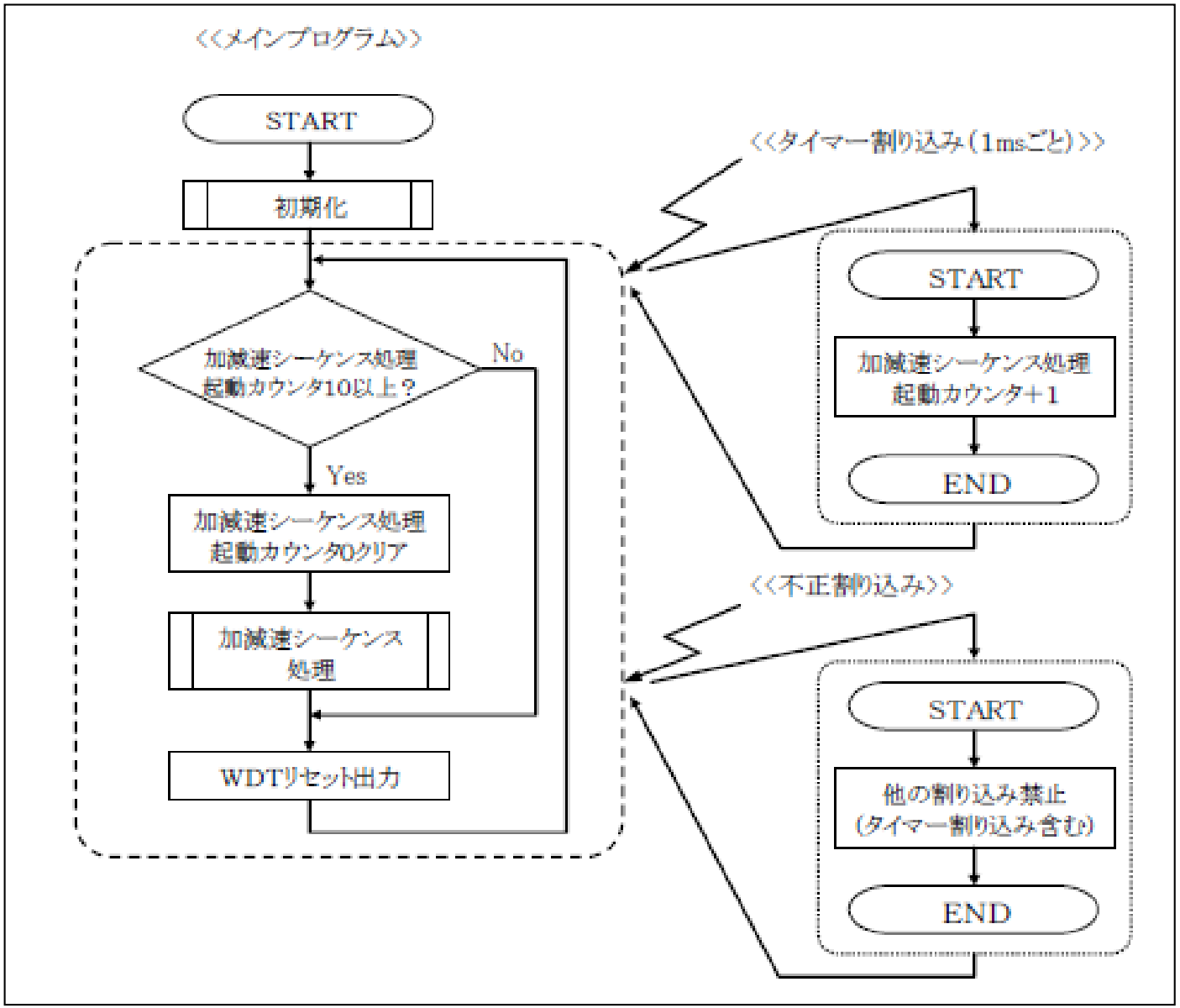
本件編成の制御回路はマスコンによるマニュアル操作で、力行は4ノッチまである。マスコンからの力行指令を受けたVVVFインバータは、加速制御のシーケンスをソフトウェアで処理してトルク指令を出し、力行ノッチに応じたトルクを発生させ、列車を加速させる。

加速制御のシーケンスには戻しノッチ機能がないため、いったん投入された力行指令は高位（数字の多い位置）の指令があるか、又は、ノッチオフされるまで保持される。

「戻しノッチ機能」とは、高位のノッチから低位のノッチに戻すと、直接戻した低位のノッチ指令に移行する機能をいう。

© 2015 IPA, All Rights Reserved

システム構造把握の具体例③　加減速制御プログラム



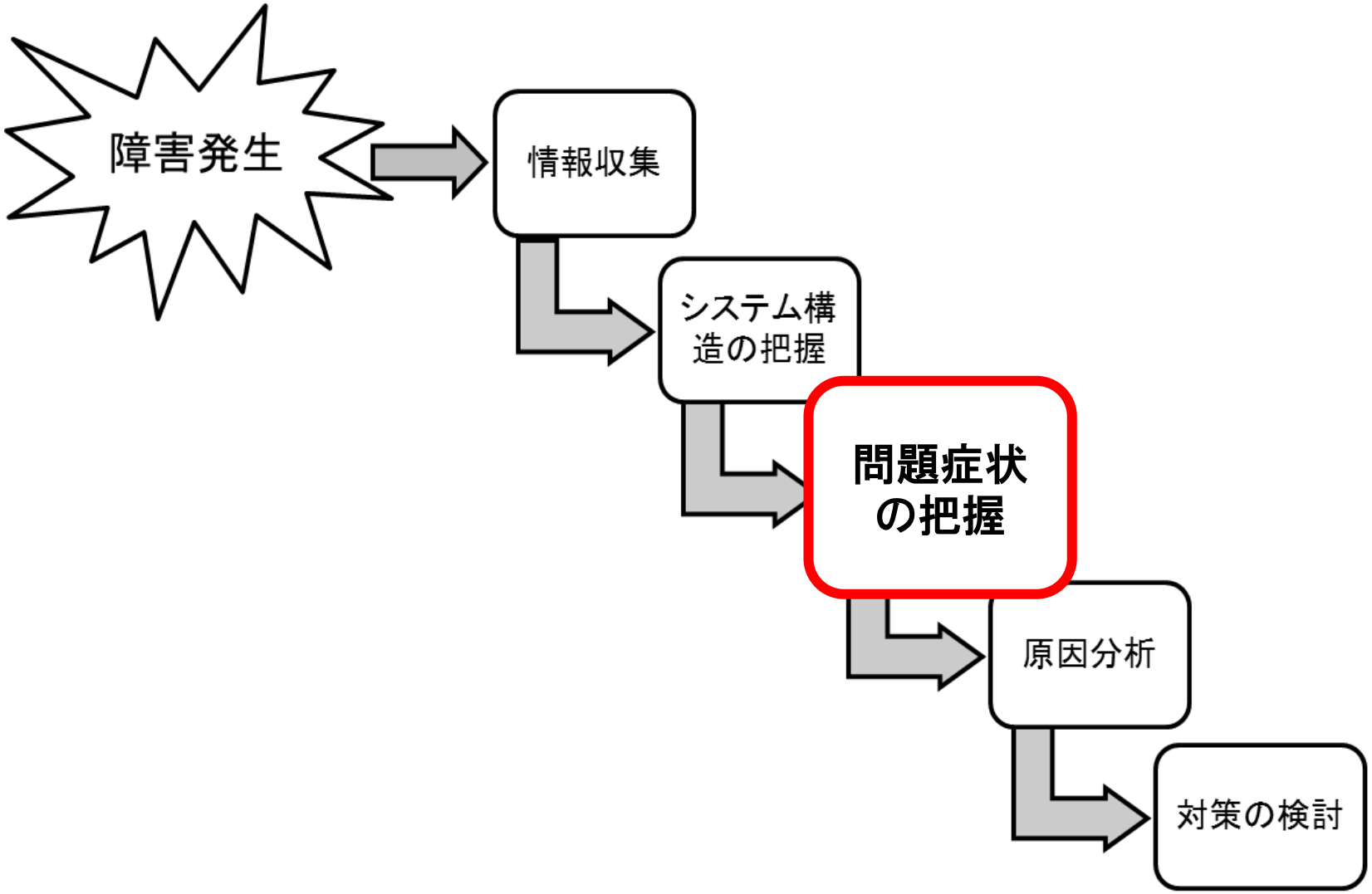
© 2015 IPA, All Rights Reserved

システム構造把握の具体例④　ブレーキシステム

常用ブレーキは、回生ブレーキと空気ブレーキを併用している。主として回生ブレーキが作用するが、回生電流が立ち上がっていないブレーキ開始時と停止直前には空気ブレーキが作用する。
回生ブレーキ作用中は、運転台右側にある表示灯群の中にある黄色の「電制」表示灯が点灯する。
各運転台には、回生ブレーキの開放用のスイッチとして‘「電制ブレーキ」スイッチ’（以下「電制スイッチ」という。）が設けられている。このスイッチを「切」とすると、常用ブレーキにおいても回生ブレーキは作用せず、常に空気ブレーキのみが作用する。
非常及び保安ブレーキでは、空気ブレーキのみが作用する。
ブレーキ指令はすべて電気指令となっている。常用及び保安ブレーキは指令線の加圧で動作し、非常ブレーキは常時加圧された指令線が無加圧となることで動作する。
保安ブレーキを動作させるための圧力空気は、「保安だめ」と呼ばれる空気だめから常用及び非常ブレーキとは別の配管を経由して供給され、常用及び非常ブレーキの空気系統に異常があった場合にバックアップする仕組みとなっている。

© 2015 IPA, All Rights Reserved

3. 問題症状の把握



© 2015 IPA, All Rights Reserved

問題症状把握の手法

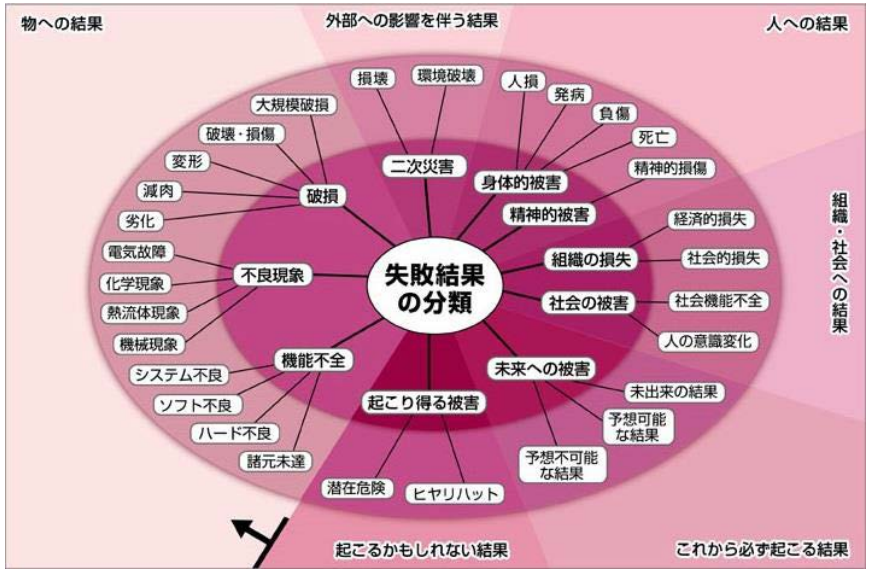
タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	**問題症状の把握**	**事故経過表、VTA**
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

© 2015 IPA, All Rights Reserved

問題症状の把握

収集した情報の中から、問題と思われる事象（結果）を抜き出し、問題症状を把握する。

問題事象の抽出には「失敗まんだら（結果まんだら）」を観点として用いるとよい。



図．失敗まんだら（結果まんだら）
【引用】畑村洋太郎, 失敗知識データベースの構造と表現, JST (2005)
http://www.sozogaku.com/fkd/inf/mandara.html

© 2015 IPA, All Rights Reserved

問題症状把握のポイント

SEC
Software Reliability Enhancement Center

☞ 作業のポイント

- 問題症状は最終状態（結果）からことの発端となった事象まで遡っていく。
- あるべき姿や原理原則からの逸脱を洗い出していく。
- 問題解決までのスピードが求められる場合は、原因にあたりをつけて、ある程度範囲を絞って問題を把握していく。ただし、技術的知識や経験を持っていることが前提。
- 複数の問題が折り重なって障害に至ってしまったような場合は、最終状態に至るまでの途中の問題症状も把握することが重要。

© 2015 IPA, All Rights Reserved

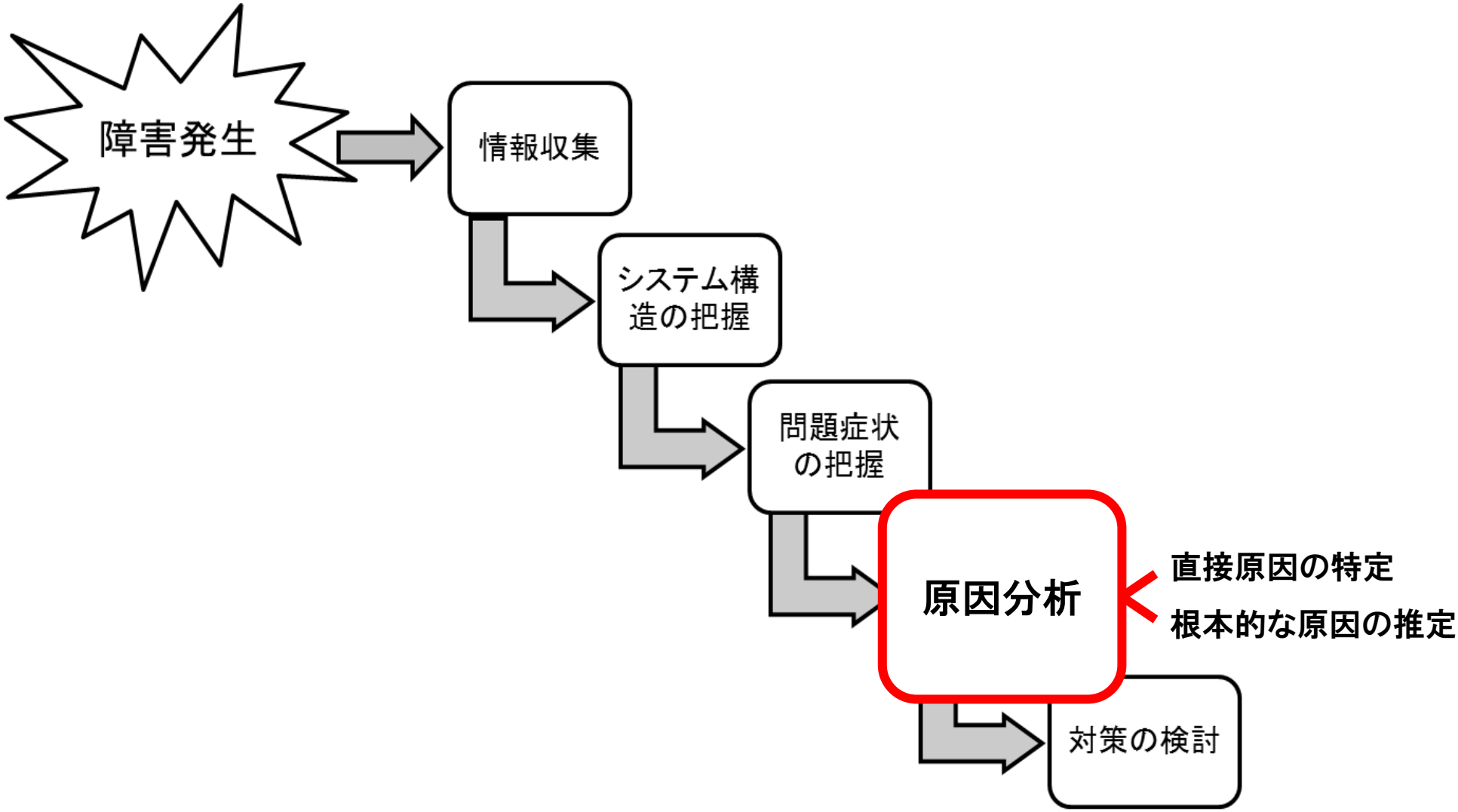
問題症状把握の具体例

運転士がワンハンドルマスコンを力行位置としていないにもかかわらず加速する状態になった。
車両の異常に気付いたが、すぐに列車を止めようとは思わなかった。最初はマスコンの入れ間違いか何かでノッチオフすれば戻るのではないかと思った。
運転士は非常ブレーキ及び保安ブレーキを使用したが所定位置に停止しなかった。
下り列車は、停車駅の所定位置を通り過ぎ、その先の分岐器に衝突、接近していた上り普通列車の進路を支障して停止した。

本事例では、“コト”を意識することによって、運転手の思い込みによって判断が遅れていたという問題点を把握することができた。

© 2015 IPA, All Rights Reserved

4. 原因分析



© 2015 IPA, All Rights Reserved

原因分析手法

SEC Software Reliability Enhancement Center

タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	問題症状の把握	事故経過表、VTA
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

© 2015 IPA, All Rights Reserved

原因分析

問題症状を引き起こした直接原因(モノの視点)を特定する。また、問題症状を引き起こした原因行動(ヒト、コトの視点)を分析する。行動分析には「失敗まんだら(行動まんだら)」を観点として用いるとよい。

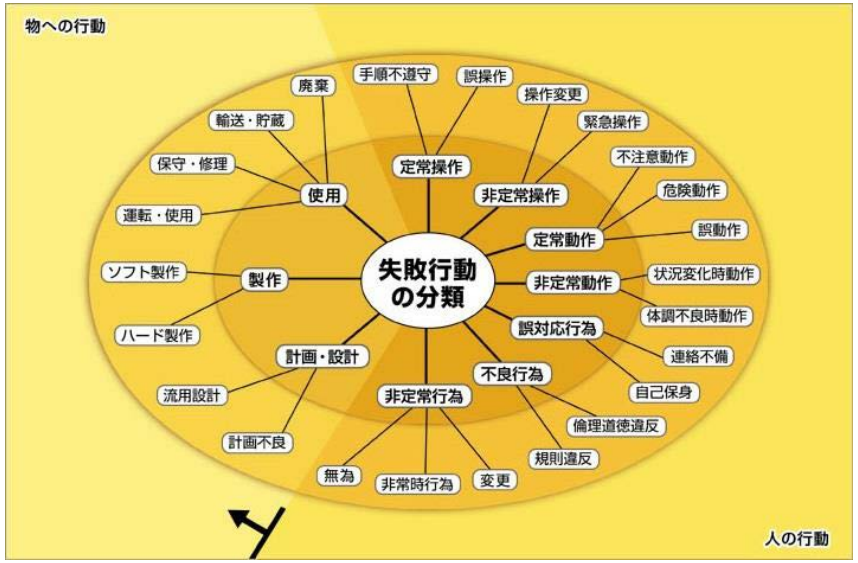


図. 失敗まんだら(行動まんだら)
【引用】畑村洋太郎, 失敗知識データベースの構造と表現, JST (2005)
http://www.sozogaku.com/fkd/inf/mandara.html

© 2015 IPA, All Rights Reserved

☞ 作業のポイント

- 組み込みシステムではハードウェア(HW)要因から探っていく。
  - 開発終盤以降はHWが原因であることが経験的に多かった。
  - HWの修正は時間がかかるので早めに見つけたほうがよい。
  - HWが原因であっても「ソフトウェアで何とかして」と言われることがある。
- 怪しそうなモジュールを抽出しテストしてみる。
  - 静的解析や単体テストだけでは不十分。結合して動かした際のI/Fや状態遷移において不具合が検出されることも多い。
- どうしても原因が見つからない場合は、マイコン自体やコンパイラ、ミドルウェアなどの不具合も疑ってみる。
- ベテラン技術者は経験や知識で怪しそうなところを特定している（鼻が利く）。若手技術者は教訓集などを参考にするとよい。

© 2015 IPA, All Rights Reserved

原因分析の具体例

２台あるVVVFインバータのうちの１台が、誤動作により力行継続状態となった。
運転士が列車の異常に気付きながら運転を継続した。 本事例ではシステムの一部に「運転手」（ヒト）が含まれていたため、「行動まんだら」による問題行動分析を行っている。
低圧回路のマイナス極側に重畳したノイズの影響を受けやすい状態となっていた。
未使用のモニタ伝送回路に対して適切なノイズ対策がなされていなかった
ウォッチドッグタイマによる保護動作が働かなかった。

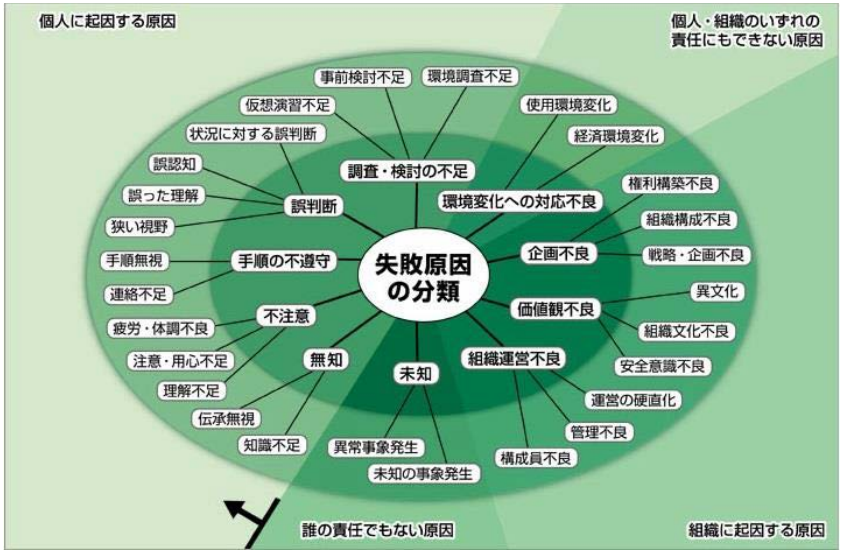
© 2015 IPA, All Rights Reserved

根本的な原因の推定

SEC
Software Reliability Enhancement Center

個人の「責任追及」ではなく、あくまで「仕事の改善」による再発防止、未然防止を目的として、原因が何に起因するかの観点で進める。

対策を考えやすくするため、根本原因は「失敗まんだら（原因まんだら）」の観点で整理するとよい。



図．失敗まんだら（原因まんだら）
【引用】畑村洋太郎, 失敗知識データベースの構造と表現, JST (2005)
http://www.sozogaku.com/fkd/inf/mandara.html

© 2015 IPA, All Rights Reserved

根本原因推定のポイント

SEC
Software Reliability Enhancement Center

☞ 作業のポイント

- 問題症状からスタートして、”コト“（仕事のやり方やそうしてしまった理由）を中心に深堀りをしていく。
- 仕事の3要素（技術、プロセス、マネジメント）の観点で原因を探っていく。
- プロセスの観点では、さらに「作りこみ要因」と「流出要因」の2つの観点で原因を探る。
- マネジメントの観点では、プロジェクトの目的設定の妥当性やプロジェクトの置かれた状況なども考慮に入れる。

© 2015 IPA, All Rights Reserved

根本原因推定の具体例　なぜなぜ分析

SEC Software Reliability Enhancement Center

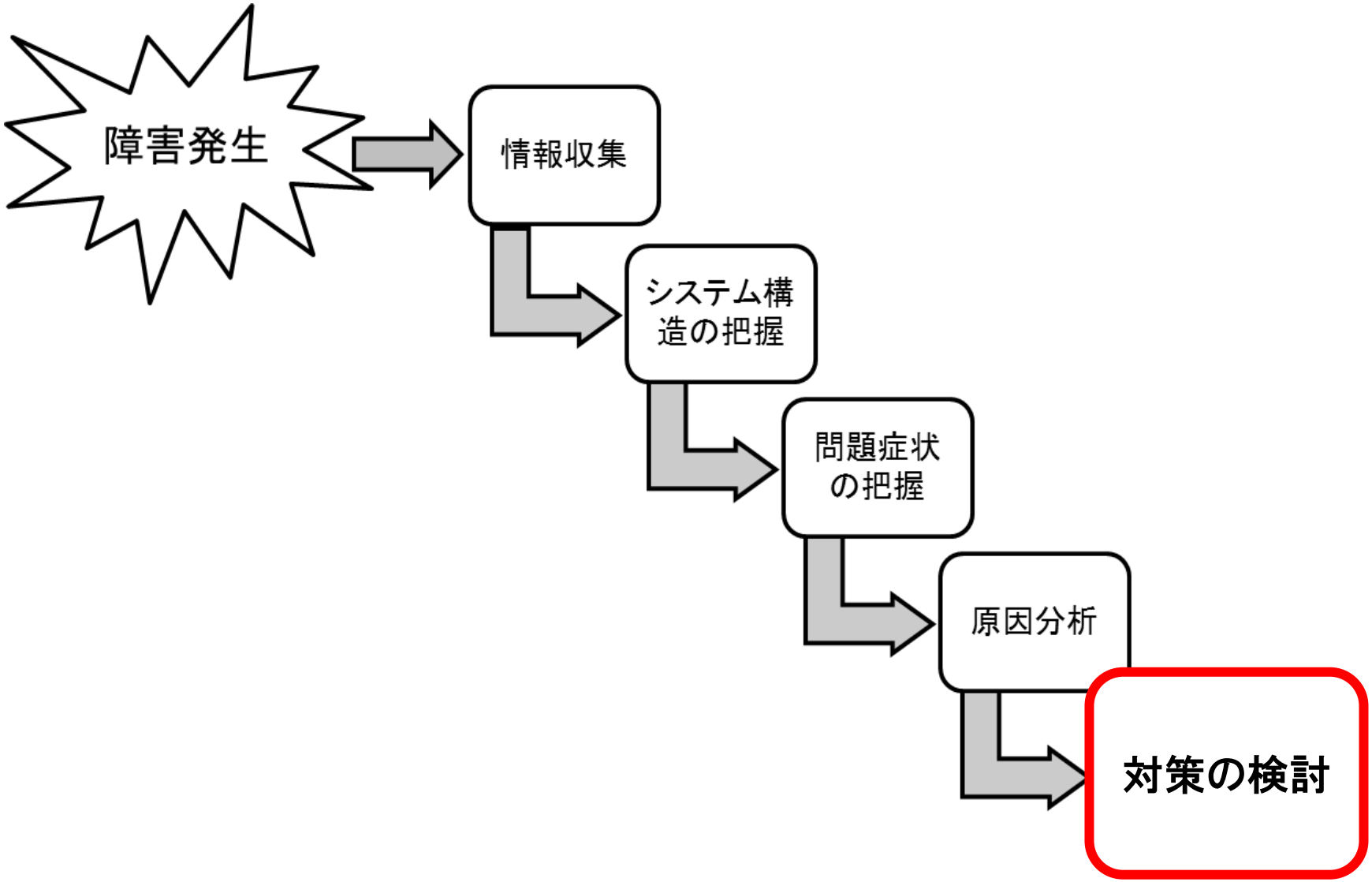
なぜなぜ分析

事象	要因				
	一次	二次	三次	四次	五次
2台あるVVVFインバータのうちの1台が、誤動作により力行継続状態となった。	搭載されていない運転台モニタの伝送回路からの割り込みを有効な割り込みとして処理するプログラムになっていたので、不正割り込みが発生した。	運転台モニタが未接続のまま使用されると知らずに、未使用の端子を有効なままにした。	運転台モニタが未接続の状態で使用されるとは知らなかった。	VVVFインバータに対する要求仕様が不明確だった。	
	不正割り込みにより、すべての割り込みが禁止され、加減速シーケンス処理が実行されなくなった	割り込み禁止としたときに、他の割り込みへ与える影響を検討できていなかった。			
		不正割り込みが発生したときの振る舞いについて、十分に検証が行われていない。	正常系は設計時に十分に検討したが、異常系の検討が漏れていた。		
			異常系に配慮したテストを行っていなかった。		
		1ms 間隔のタイマ割り込みを禁止すれば、WDTの保護動作が働くと思っていた。	タイマ割り込みを禁止したときに、期待通りにWDTの保護動作が働くことをテストで確認していなかった。		
	プログラムがデッドロックしても、ウォッチドッグタイマによる保護動作が働かなかった	加減速シーケンス処理が実行されない場合でもWDTがかからないような処理フローとなっている。	加減速シーケンス処理起動カウンタ値の判定結果にかかわらず、必ず通るパス上にWDTリセット処理があった。	WDTの利用についてのノウハウが不足していた。	

根本原因の推定では、「なぜなぜ分析」を用いることが多い

© 2015 IPA, All Rights Reserved

5. 対策の検討



© 2015 IPA, All Rights Reserved

対策検討手法

SEC
Software Reliability Enhancement Center

タスク	作業	手法
情報収集	情報収集と整理	障害管理表の調査、関係者へのヒアリング、実地調査 ログ解析
システム構造の把握	システム構造の整理	ブロック図、UML、SysML、ネットワーク図
問題症状の把握	問題症状の把握	事故経過表、VTA
原因分析	原因箇所の特定	問題行動分析、例外分析、FTA/FMEA
	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	レビュー、コードインスペクション、再試験、シミュレーション、モデル検査
	根本的な原因の推定	PNA、発生源・検出漏れ分析、なぜなぜ分析、KJ法
対策の検討	根本対策(再発防止策)の検討	影響レベル評価尺度・評価手法

© 2015 IPA, All Rights Reserved

対策の検討

SEC
Software Reliability Enhancement Center

なぜなぜ分析によって特定された根本原因について、「仕事の改善」の視点から再発防止策を検討する。また、それぞれの再発防止策について対策者を明確にし、確実に実行されるようにする。

■仕事の改善

- 技術知識・スキルの獲得、継承、向上
- プロセス改善
- マネジメント改善

© 2015 IPA, All Rights Reserved

対策検討のポイント

☞ 作業のポイント

- 障害の「結果」ー「行動」ー「原因」をまとめて構造化し、障害発生のメカニズムを一目でわかるようにしておくと対策を考えやすい。
- プロセスの改善を図る場合は、作りこみ要因(上流工程)の改善を優先する。
- 重度のチェックリスト依存は形骸化を招く。
- IPA/SECの成果物をうまく活用する。
  - 技術知識・スキル・・・ESDR、ESTRから改善のヒントを得る。
  - プロセス・・・ESPRから改善のヒントを得る。
  - マネジメント・・・ESMR/ESMGから改善のヒントを得る。

© 2015 IPA, All Rights Reserved

対策検討の具体例

SEC
Software Reliability Enhancement Center

問題の症状		問題行動		根本原因		対策案		
分類	内容	分類	内容	分類	内容	対策方法	対策者	備考
不良現象	VVVFインバータ内のゲート電源装置の高周波ノイズが、同装置の電源マイナス極側である低圧車体接地線に重畳した。	ハード制作	マイナス極と車体間の接地線の断面積が小さく、抵抗値が大きい					
		ハード制作	低圧車体接地線の配線の引き回しが長い					
不良現象	未使用のモニタ伝送回路がノイズの影響により、異常動作した	ハード制作	VVVFインバータ内の分圧抵抗盤で終端処理がされていなかった。					
機能不全	加減速シーケンス処理が実行されなくなり、力行継続状態となった	ソフト制作	搭載されていない運転台モニタの伝送回路からの割り込みを有効な割り込みとして処理するプログラムになっていたので、不正割り込みが発生した。	環境調査不足	運転台モニタが未接続のまま使用されると知らずに、未使用の端子を有効なままにした。	開発に着手する前に、システム要求定義プロセスを実施し、VVVFインバータに対する要求を明確にする。	VVVFインバータメーカー	組み込みソフトウェアの開発プロセスについては、ESPR Ver2.0を参照
		ソフト制作	不正割り込みにより、すべての割り込みが禁止されたため、加減速シーケンス処理起動カウンタの値が更新されなくなった。	調査・検討の不足	不正割り込みが発生したときの振る舞いについて、十分に検証が行われていない。	ソフトウェアを設計する際は、正常系を設計すると同時に異常系も併せて設計する。	制御プログラム作成者	ESDR p.22「A-2」を参照
				調査・検討の不足	すべての割り込みを禁止としたときに、他の割り込みへ与える影響を検討できていなかった。	ソフトウェアの詳細な設計を行う前に、アーキテクチャ(振る舞いと構造)について十分検討し、「アーキテクチャ設計書」として文書化する。アーキテクチャ設計の妥当性を確認するために、有識者を交えてレビューを実施する。	制御プログラム作成者	「ソフトウェア・アーキテクチャ設計」プロセスについては、ESPR Ver2.0 p.76 を参照。
				誤判断	タイマ割り込みを禁止すれば、WDTの保護動作が働くと誤認識していた。	「設計はできる限り単純化し、テストしやすさを考慮して設計する。」といったような、設計基準を設ける。	制御プログラム作成者	ESDR p.28「A-6」を参照。
				手順の不遵守	タイマ割り込みを禁止したときに、期待通りにWDTの保護動作が働くことをテストで確認していなかった。	フェイルセーフを実装した場合は、実装したフェイルセーフが期待通りに動作することを必ずテストで確認する。	制御プログラム作成者	
				手順の不遵守	ソフトウェアに対する安全要求が不明確だった。	ソフトウェア開発に着手する前に、安全要求定義プロセスを実施し、ソフトウェアの機能に対する要求を明確にする。	制御プログラム作成者	「安全要求定義」プロセスについては、ESPR Ver2.0 p.132 を参照。
				価値観不良	加減速制御プログラムが「システムの安全にかかわる重要な要素である」という認識が欠けていた。	機能安全の「セーフティインテグリティレベル」の考え方を取り入れ、システムの安全性に関わる部品かどうかを判断する仕組みを構築する。	VVVFインバータメーカー 車両メーカー	機能安全については、IEC61508などの国際規格を参照。

© 2015 IPA, All Rights Reserved

まとめ

- 失敗から学ぶ手法(障害分析手法)を習得しましょう。
- ノウハウを獲得する仕組みを整備して継続実行しましょう。
- ノウハウを使える形にして、共有・伝承していきましょう。

© 2015 IPA, All Rights Reserved

ご清聴ありがとうございました

© 2015 IPA, All Rights Reserved